(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 March 2002 (07.03.2002)

PCT

(10) International Publication Number
WO 02/19661 A2

(51) International Patent Classification[7]: H04L 29/06

(21) International Application Number: PCT/US01/41961

(22) International Filing Date: 30 August 2001 (30.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/653,045 1 September 2000 (01.09.2000) US

(71) Applicant: **TOP LAYER NETWORKS, INC.** [US/US]; 2400 Computer Drive, Westboro, MA 01581-1770 (US).

(72) Inventors: **NARAYANASWAMY, Krishna**; c/o Top Layer Networks, Inc., 2400 Computer Drive, Westboro, MA 01581-1770 (US). **SPINNEY, Barry, A.**; c/o Top Layer Networks, Inc., 2400 Computer Drive, Westboro, MA 01581-1770 (US). **ROSS, Theodore, L.**; c/o Top Layer Networks, Inc., 2400 Computer Drive, Westboro, MA 01581-1770 (US). **PAQUETTE, Michael, D.**; Top Layer Networks, Inc., 2400 Computer Drive, Westboro, MA 01581-1770 (US). **WRIGHT, Christopher, L.**; Top Layer Networks, Inc., 2400 Computer Drive, Westboro, MA 01581-1770 (US).

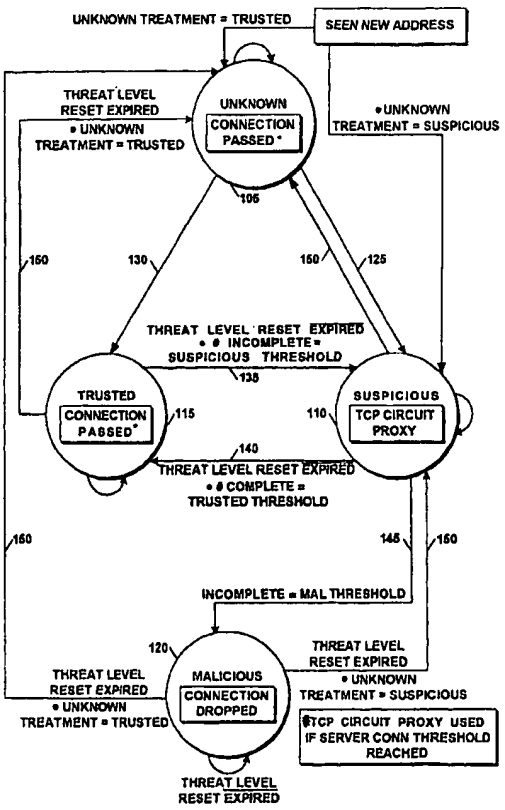
(74) Agents: **CHOW, Stephen, Y.** et al.; Perkins, Smith & Cohen, LLP, One Beacon Street, Boston, MA 02108 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: SYSTEM AND PROCESS FOR DEFENDING AGAINST DENIAL OF SERVICE ATTACKS ON NETWORK NODES



(57) Abstract: The present invention is a network switch that maintains a relatively lightly loaded state, and at the same time protects the network servers from DOS and DDOS attacks. The switch maintains a very large table of IP addresses where it stores information such as the number of incompleted and completed connections from each address. Using this information, the switch classifies each address into a threat level: unknown, trusted, suspicious, and malicious. Each threat level is treated differently allowing the switch to provide efficient access to the server while maintaining security. Connection to the server is denied to clients classified as malicious while trusted clients are passed through to the server. Suspicious connections are proxied while unknown connection treatment may be set by the user.

WO 02/19661 A2

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND PROCESS FOR DEFENDING AGAINST DENIAL OF SERVICE ATTACKS ON NETWORK NODES

FIELD OF THE INVENTION

This invention relates generally to computer security and more particularly to preventing "denial of service" attacks through SYN flooding of a server connected to a computer network using the TCP/IP protocol.

BACKGROUND OF THE INVENTION

Networks and the computers connected by networks have many vulnerabilities. One vulnerability is susceptibility to Denial of Service (DOS) attacks and Distributed Denial of Service (DDOS) attacks. These problems have recently brought down many prominent web sites and rendered them useless for many hours.

A common type of DOS attack is called SYN flooding. Typically, a TCP-based application connection is set up by a 3-way handshake that is completed between a client and server. The client asks for a connection to the server by sending a SYN (synchronization) message to the server. In response, the server allocates resources and responds to the client with a SYN-ACK (synchronization acknowledgment) message. The client, upon receiving the SYN-ACK, replies to the server with an ACK, and the connection then enters the ESTABLISHED state.

If the 3-way handshake does not complete normally, the server generally holds on to the state for as long as 2 minutes before releasing the connection and the allocated resources. The DOS and DDOS attacks involve rogue client machines initiating multiple connections that do not respond to the SYN-ACK sent by the server. The server soon runs out of resources and cannot set up new connections from legitimate clients. Under certain circumstances, the servers must be rebooted before they can return to normal behavior. If the attack persists, the server could be taken out of service for a long time.

Another type of DOS attack involves repeated complete TCP connections. For example, repeated HTTP GET requests from an

Internet client to a server. While the HTTP GET command is small, the amount of data requested from the server can be large. With each HTTP GET command potentially using up a significant percentage of the server resources, the server can be potentially overwhelmed. A large number of HTTP GET requests could overwhelm a server even without requesting large amounts of data. Further examples of potential attacks are FTP GET messages and SMTP (e-mail) SEND commands.

There are other solutions in the current art that attempt to solve the DOS and DDOS attack problems. The most popular solutions involve a third device (usually a switch or router) acting on behalf of the server, known as a "proxy," until the 3-way handshake completes successfully. The proxy then connects to the server. This is a resource-intensive process. Current art proxy devices have difficulty handling even a reasonable network loadIt remains desirable to protect network servers without sacrificing server availability.

It is an object of the present invention to provide a method and apparatus to reduce vulnerability of networked servers to DOS and DDOS attacks.

SUMMARY OF THE INVENTION

The problems of Denial of Service and Distributed Denial of Service attacks in a computer network are solved by the present invention of a SYN flood mitigation system and process.

The present invention is a network switch that maintains a relatively lightly loaded state, and at the same time, protects the network servers from DOS and DDOS attacks. The switch maintains a very large table of IP Addresses (for example ~200K,, however larger tables are possible within the scope of the present invention) where it stores information such as the number of incomplete/completed connections from each address. Using this information, the switch classifies each address into a threat level: unknown, trusted, suspicious and malicious. Each threat level is treated differently allowing the switch to provide efficient access to the server while maintaining security. Connection to the server is

denied to clients classified as malicious, while trusted clients are passed through to the server. Suspicious connections are proxied, while unknown connection treatment may be set by the user.

By continuously keeping track of how many connection attempts are made by each client and how many of those are completed successfully, the switch transitions a client's IP address into one of the above mentioned Threat Levels and treats the attempted connection to the server accordingly.

The present invention together with the above and other advantages may best be understood from the following detailed description of the embodiments of the invention illustrated in the drawings, wherein:

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a web site having a switch according to principles of the invention connected to the Internet;

Figure 2 is a schematic diagram of communications between a client and a server through the switch of Figure 1;

Figure 3 is a state diagram of the switch of Figure 1;

Figure 4 is a first flow chart of the operation of the switch of Figure 1;

Figure 5 is a second flow chart of the operation of the switch of Figure 1 when a request for connection from a trusted IP address is received;

Figure 6 is a third flow chart of the operation of the switch of Figure 1 when a request for connection from a suspicious IP address is received; and

Figure 7 is a fourth flow chart of the operation of the switch of Figure 1 when a request for connection from an unknown address is received.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Figure 1 is a block diagram of the switch according to principles of the present invention. A web site 10 connected to the Internet 15 has a router 20 connected to a switch 25 operating according to the present invention. The switch 25

has a plurality of servers 30 and a database 100 connected to it. A plurality of clients 35 connected to the Internet 15 may access the servers 30 on the Web site 10 through the switch 25. The servers may be accessed over the Internet using the TCP/IP protocol.

Figure 2 is a schematic diagram of establishing a proxied connection through the switch 25 of the present invention from a client 50 to a server 55.

Figure 2a shows an incomplete connection from the client 50 to the server 55. In attempting to establish a connection to the server 55, the client 50 sends a SYN message 60 which is received by the switch 25. The switch 25 allocates resources for a connection and sends a SYN-ACK message 65 to the client 50. The switch 25 then waits for a period of time for an ACK message from the client 50. The period of time may be, for example, 15 seconds, but it may be longer or shorter within the scope of the present invention. If, after the waiting period, the switch 25 does not receive an ACK message from the client 50, the switch releases the resources.

Figure 2b shows a completed proxied connection from the client 50 to the server 55 through the switch 25. The client 50 sends a SYN message 60. The switch 25 allocates resources and responds with a SYN-ACK message 65. The client 50 replies to the SYN-ACK message 65 with an ACK message 70 to complete a connection to the switch. When the ACK message 70 is received from the client, the switch sends a SYN message 75 to the server 55. The server 55 responds by allocating resources and sends a SYN-ACK message 80 to the switch 25. The switch responds with an ACK message 85 to complete the connection to the server 55 and the proxy connection to the client 50.

Figure 2c shows the switch 25 providing a data connection between the client 50 and the server 55. The switch 25 provides splices 90, 95 to match the client data sequence number with the server data sequence number.

Figure 3 shows a state diagram of the switch of the present invention. The switch classifies a client attempting to connect to a server as one of four types: unknown 105, suspicious 110, trusted 115 or malicious 120. When the client

attempts to connect to the server, a client identifier and the connection attempt are noted as described below. In the present invention, the client identifier is the IP address of the client. The identifier of the client and the associated client type are stored in a large database 100 (Figure 1) which the switch consults whenever a client requests a connection to a server connected to the switch. The database may be in the switch itself or it may be remote from the switch. The server may be directly or indirectly connected to the switch. Two counts are kept for each IP address in the database, a completed connections count and an incomplete connections count. The switch maintains a threshold value called the trusted threshold, which is associated with the completed connections count. The switch has two additional thresholds, the suspicious threshold and the malicious threshold, which are associated with the incomplete connections count.

A detailed description of the classification is as follows.

1) *Unknown* 105 - If a client is connecting for the first time, or after a sufficient length of time has passed since the client's previous connection, the client is classified as unknown. The switch may be set to treat an unknown client as either suspicious or trusted.

2) *Suspicious* 110 - If the number of incomplete connection setups from a particular client exceeds the suspicious threshold, the client is classified as suspicious.

3) *Trusted* 115 - If the number of successfully completed connections from a particular client is greater than a trusted threshold, the client is classified as trusted.

4) *Malicious* 120 - If the number of incomplete connection setups from a particular client exceeds the malicious threshold, then the client is classified as malicious.

Any client IP address that is classified as trusted is considered good and its connection request is sent to the server as normal. The user of the switch may also set the switch so that client IP addresses that are unknown may be treated as trusted.

Any connection request from a client IP address that is in the Suspicious state is always proxied by the switch. The server is contacted only upon successful completion of a connection between the client and the switch.

Any connection request from a client IP address that is in the malicious state is discarded.

The state transitions are as follows:

Unknown ---> **Suspicious** 125 A client is reclassified as suspicious from the unknown state when the number of incomplete connection requests is greater than or equal to the suspicious threshold of the server connect threshold.

Unknown ---> **Trusted** 130 A client is reclassified as trusted from the unknown state when the number of completed connection requests is greater than or equal to the trusted threshold.

Trusted ---> **Suspicious** 135 A client is reclassified as suspicious from a trusted state when the number of incomplete connection requests is greater than or equal to the suspicious threshold.

Suspicious ---> **Trusted** 140 A client is reclassified as suspicious from the trusted state when the number of completed connection requests is greater than or equal to the trusted threshold.

Suspicious ---> **Malicious** 145 A client is reclassified as malicious from the suspicious state when the number of incomplete connection requests is greater than or equal to the malicious threshold.

The initial state assigned to an address is based on the unknown treatment parameter set by the user. If it is set to suspicious, the initial state of an unknown client is suspicious and the first connection request is proxied. Otherwise the initial state is left classified as unknown which is treated in a manner similar to the trusted state.

In any one of these states, if there are no new connection requests for a period of time set in a ThreatLevelReset Time value, the address is returned to an initial state of unknown 150, except in the case where the transition is from the malicious state and the initial value for unknown clients is suspicious.

Figure 4 is a first flow chart of the operation of the switch of the present invention. When a client requests a connection to a server connected to the switch, the switch receives a SYN message from the client which includes the client's IP address, block 200. The switch checks the table of IP addresses and associated types, block 205. If the client IP address is not found in the table, then the client is unknown. The IP address is added to the table and the switch treatment parameter is set, block 210. If the client IP address is found in the table, the switch then checks the table for the last time the client requested a connection, block 215. If the period of time since the last connection request exceeds the time limit, then the treatment parameter for the client is reset to unknown, block 215.

Then the switch increments an incomplete connections counter associated with the IP address, block 220. The switch then reads the treatment parameter for the client from the table, block 225, and then moves to that state. If the client is of type malicious, the packet from the client is dropped and no connection is established, block 230.

If the client is trusted, the client is handled according to Method A 235 shown Figure 5. If the client is suspicious, the client is handled according to Method B 240 as shown in Figure 6. If the client is unknown, the client is handled according to Method C 245 in Figure 7.

Referring now to Figure 5, when the client is a trusted client, the switch determines if the number of incomplete connections is greater than the suspicious threshold 250. If the suspicious threshold is exceeded, the client's status is changed to suspicious 255. The switch resets the completed connections counter for the client. The switch then establishes a proxy connection between the server and the client 260.

If the suspicious threshold has not been exceeded, the client remains classified as a trusted client. A trusted client requires no proxy and the connection is passed to the server, block 265. The switch then determines whether the connection between the client and the server was completed,

block 270. If the connection is completed, the completed connections counter is incremented and the incompleted connections counter is decremented, block 275. If the connection is not completed, the counters are not changed block 280.

Referring now to Figure 6, when a client is a suspicious client, the switch first determines whether the number of incomplete connections is equal to the malicious threshold, block 300. If it is, the packet is dropped and the connection is not completed, block 305. If the incomplete connections count is not equal to the malicious threshold, the switch determines whether the number of completed connections is greater than the trusted threshold, block 310. If the trusted threshold has been exceeded, the switch passes the connection through to the server, block 315. If the trusted threshold has not been exceeded, the switch acts as a proxy device between the client and the server, block 320.

Referring now to Figure 7, when a client is an unknown client, the switch first determines whether the incomplete connections counter exceeds the suspicious threshold, block 350. If the suspicious threshold is exceeded, the treatment parameter of the client is changed to suspicious and the switch forms a proxy connection between the client and the server, block 355. If the suspicious threshold has not been exceeded, the switch then determines whether the completed connections counter is greater than the trusted threshold, block 360. If the trusted threshold is exceeded, the client is classified as a trusted client and the connection is forwarded to the server, block 365. If the trusted threshold has not been exceeded, then the client remains in the unknown state and the connection is forwarded to the server, block 370.

The switch also maintains a count of valid connections per source IP address for types of requests which may be used to attack the server. The count is kept per unit time. If a client address exceeds a threshold number of valid connections of a specified type, connection attempts beyond the threshold

number are discarded. Examples of requests are HTTP GET requests, and FTP GET requests.

There is an alternate form of SYN flood attack where there are a large number of clients, i.e. a large number of addresses, each request a connection to a server. In this type of attack, the individual addresses themselves may not exceed the suspicious or malicious thresholds, and so may not be proxied or barred from connection. The server, however, could receive a higher number of connections than it can handle. In order to protect the server form this type of attack, the switch maintains a counter of incomplete TCP connections to each server. If this number exceeds a threshold called the ServerCon threshold, the requested connection is always proxied except where the address is already classified as malicious, in which case the connection request is dropped.

Every time a new connection is attempted through the switch, the switch sends a session report to a data collector. The data collector may be close to the switch or remotely located. The report indicates the threat level that was associated with the client IP address for the attempted connection. The data may be stored for purposes such as generating system messages to users monitoring the system. The data may also be extracted for analysis or into a graphical monitoring system.

In a first alternative embodiment of the invention, instantaneous thresholds may be set. The instantaneous thresholds are set to large enough values so that ACKs that are lost to due lost packets do not trigger a change in switch state possibly resulting in denial of a client. The instantaneous threshold values are also set small enough so that genuine attacks are not allowed to create a DOS condition.

In a second alternative embodiment of the invention, thresholds are randomly varied so that network scanning tools have greater difficulty in determining if a particular switch or server is in a network.

In a third alternative embodiment of the invention, a percentage of connects from "known threat" addresses may be proxied in order to confuse network scanning tools. The percentage may be based on attack load or normal load. If the switch is under heavy attack, for example, the "known threat" addresses are all, or almost all, discarded. If the switch is under light attack, then some connections from the "known threat" addresses are proxied. This provides a confused view of the network to network scanning tools and further tends to provide a different set of "available" ports at different times, i.e. at repeated attacks.

In a fourth alternative embodiment of the invention, the malicious threshold is a ratio of the number of failed connections to the number of requests greater than a selected initialization threshold. This method, however, may enable a sophisticated threat involving mixing good requests with malicious requests resulting in a DOS failure of the server because the threat level of the source IP address was not raised to the malicious level.

It is to be understood that the above-described embodiments are simply illustrative of the principles of the invention. Various and other modifications and changes may be made by those skilled in the art which will embody the principles of the invention and fall within the spirit and scope thereof.

What is claimed is:

1. In a communication network in which connections are established in part by a reservation of node resources during connection set-up, a process for defending against malicious reservation of said resources causing denial of service, said process comprising:

a) tracking for a predetermined time period the number of requests of reservation of said node resources from a particular source;

b) determining the number of said requests from said source failing to result in a completed connection;

c) if said number of failures exceed a threshold of presumed maliciousness, then rejecting further requests from said source for a predetermined time-out-period.

2. The process of Claim 1 further comprising the step of:

b') if said number of failures exceed a threshold of suspicion, then sending through a proxy further requests from said source for a predetermined watch period.

3. The process of Claim 1 wherein said threshold is a ratio of said number of failures to said number of requests greater than a predetermined initialization threshold.

4. The process of Claim 1 further comprising the step of:

d) if the number of requests from said source failing to result in a completed connection does not exceed a threshold of suspicion and the number of requests from said source resulting in a completed connection does not exceed a threshold of presumed trustworthiness, then sending through a proxy further requests from said source for a predetermined watch period.

5. The process of Claim 4 further comprising the step of:

e) if the number of requests from said source resulting in a completed connection exceeds a threshold of presumed trustworthiness, then connecting said source to said server.

6. An apparatus for mitigating attacks on a server connected to a large network, comprising:

a database for storing a plurality of client identifiers and associated client data; and

a switch to receive as input over the network an identifier of a client requesting a connection to the server, said switch to match said received identifier with a client identifier stored in said database and to operate on said requested connection according to client data associated with a matched identifier.

7. The apparatus of claim 6 wherein said switch operates according to a user-set parameter when no match for said received identifier is found in said database.

8. The apparatus of claim 6 wherein associated client data further comprises a threat level.

9. The apparatus of claim 6 further comprising a state machine to determine a threat level of a client requesting connection from said received client identifier and associated client data stored in said database.

10. The apparatus of claim 6 wherein said associated client data further comprises a count of completed connections between the client and the server and a count of incomplete connections from the client to the server.

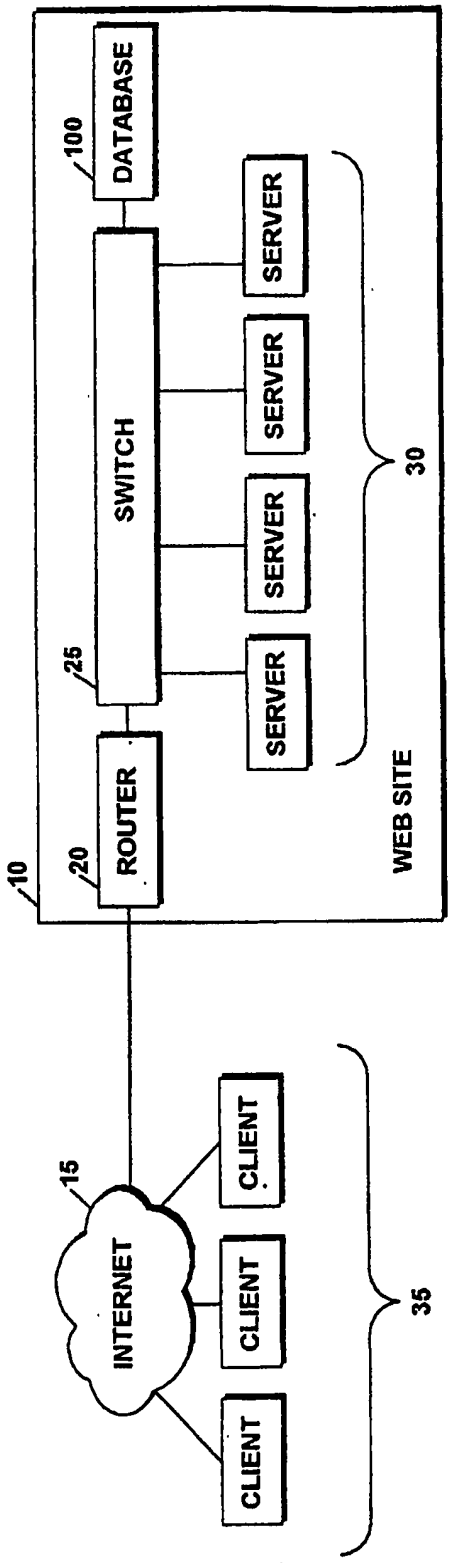
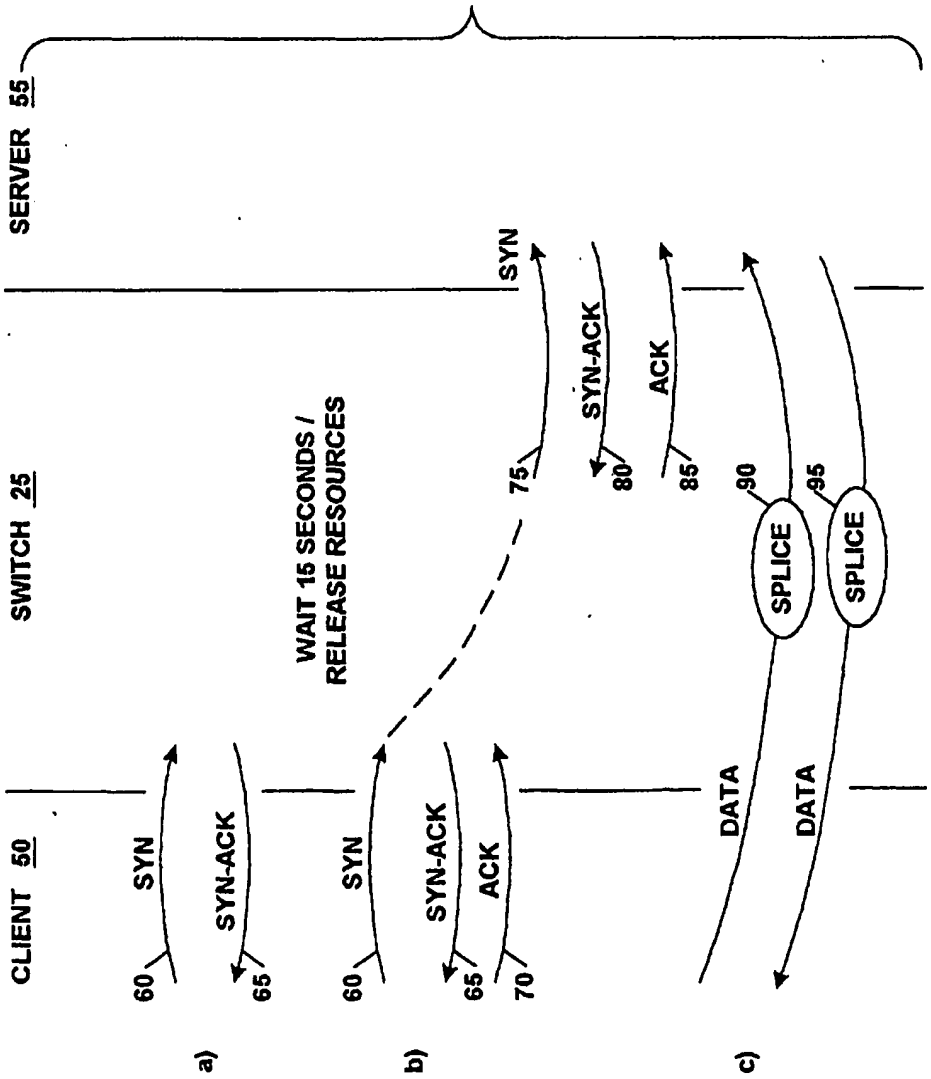

DATABASE
100
SERVER
SERVER
SERVER
SERVER
SWITCH
25
30
ROUTER
20
WEB SITE
10
INTERNET
15
CLIENT
CLIENT
CLIENT
35

Figure 1

Figure 2


CLIENT 50
SWITCH 25
SERVER 55
SYN
SYN-ACK
ACK
DATA
DATA
SPLICE
SPLICE
WAIT 15 SECONDS / RELEASE RESOURCES
60
65
70
75
80
85
90
95
a)
b)
c)

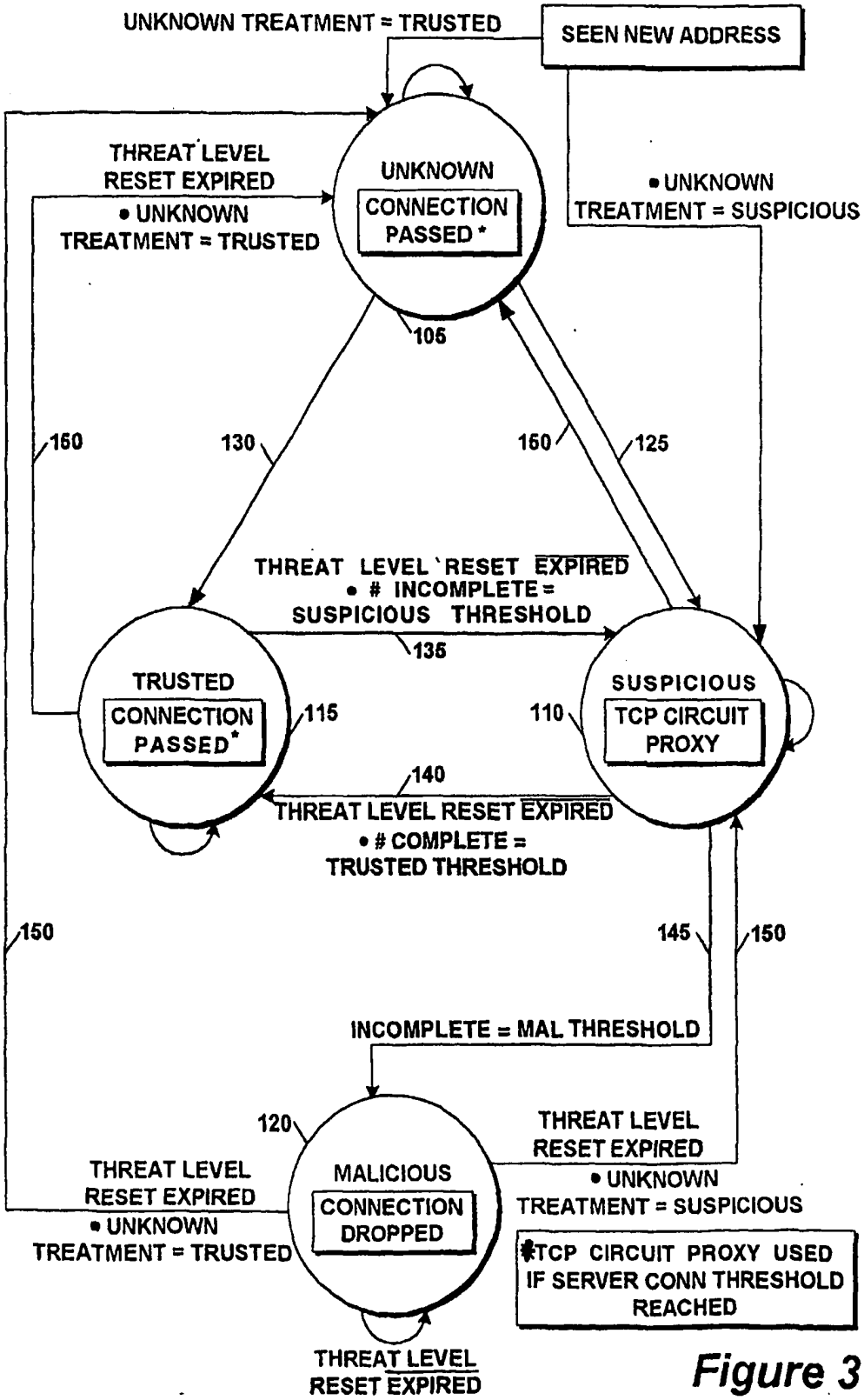

UNKNOWN TREATMENT = TRUSTED
SEEN NEW ADDRESS
THREAT LEVEL RESET EXPIRED
• UNKNOWN TREATMENT = TRUSTED
UNKNOWN
CONNECTION PASSED *
• UNKNOWN TREATMENT = SUSPICIOUS
105
160
130
160
125
THREAT LEVEL RESET EXPIRED
• # INCOMPLETE = SUSPICIOUS THRESHOLD
135
TRUSTED
CONNECTION PASSED*
115
110
SUSPICIOUS
TCP CIRCUIT PROXY
140
THREAT LEVEL RESET EXPIRED
• # COMPLETE = TRUSTED THRESHOLD
150
145
150
INCOMPLETE = MAL THRESHOLD
120
THREAT LEVEL RESET EXPIRED
• UNKNOWN TREATMENT = SUSPICIOUS
THREAT LEVEL RESET EXPIRED
• UNKNOWN TREATMENT = TRUSTED
MALICIOUS
CONNECTION DROPPED
*TCP CIRCUIT PROXY USED IF SERVER CONN THRESHOLD REACHED
THREAT LEVEL RESET EXPIRED

Figure 3

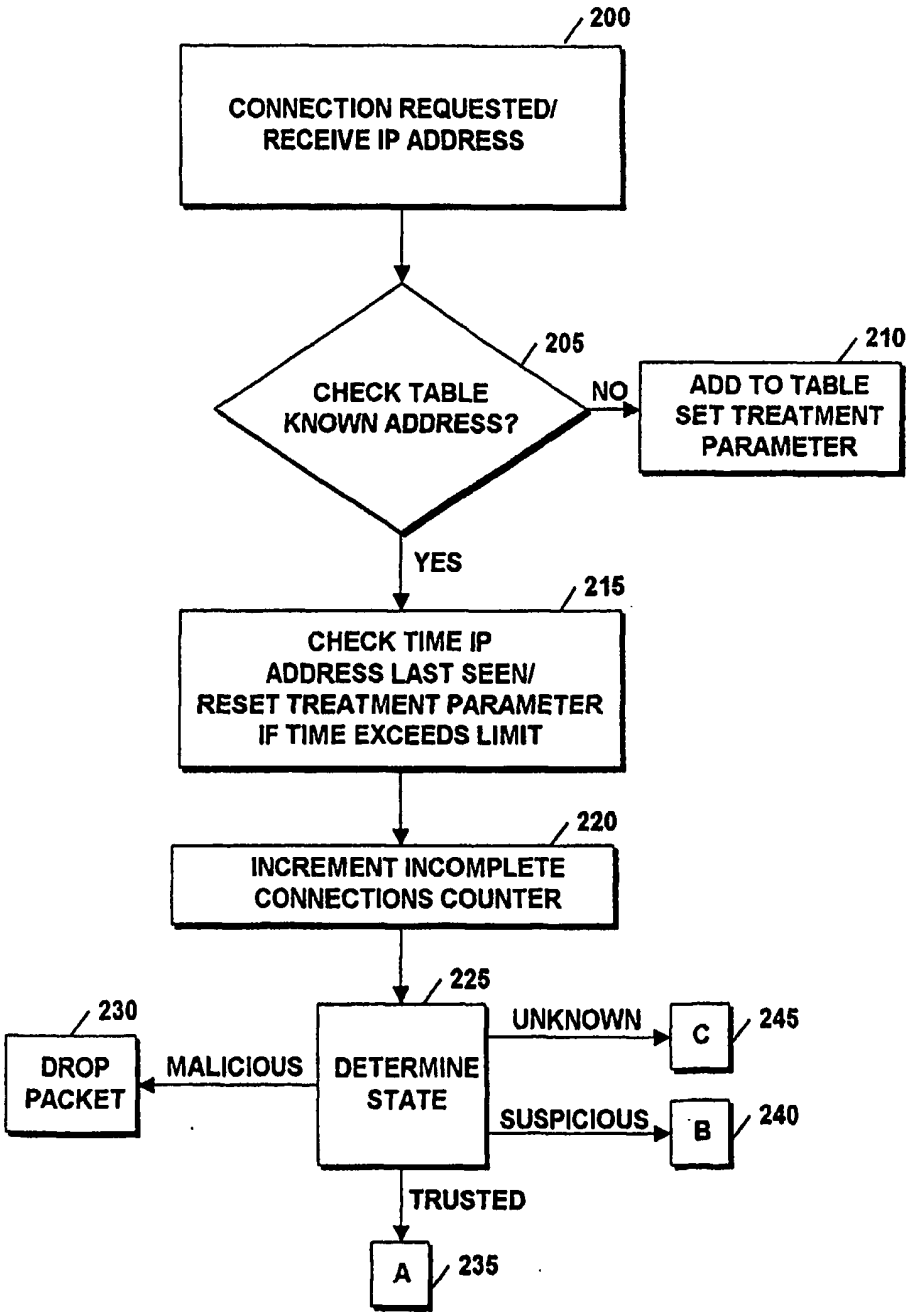

200
CONNECTION REQUESTED/
RECEIVE IP ADDRESS
205
CHECK TABLE
KNOWN ADDRESS?
NO
210
ADD TO TABLE
SET TREATMENT
PARAMETER
YES
215
CHECK TIME IP
ADDRESS LAST SEEN/
RESET TREATMENT PARAMETER
IF TIME EXCEEDS LIMIT
220
INCREMENT INCOMPLETE
CONNECTIONS COUNTER
225
230
DROP
PACKET
MALICIOUS
DETERMINE
STATE
UNKNOWN
C
245
SUSPICIOUS
B
240
TRUSTED
A
235

Figure 4

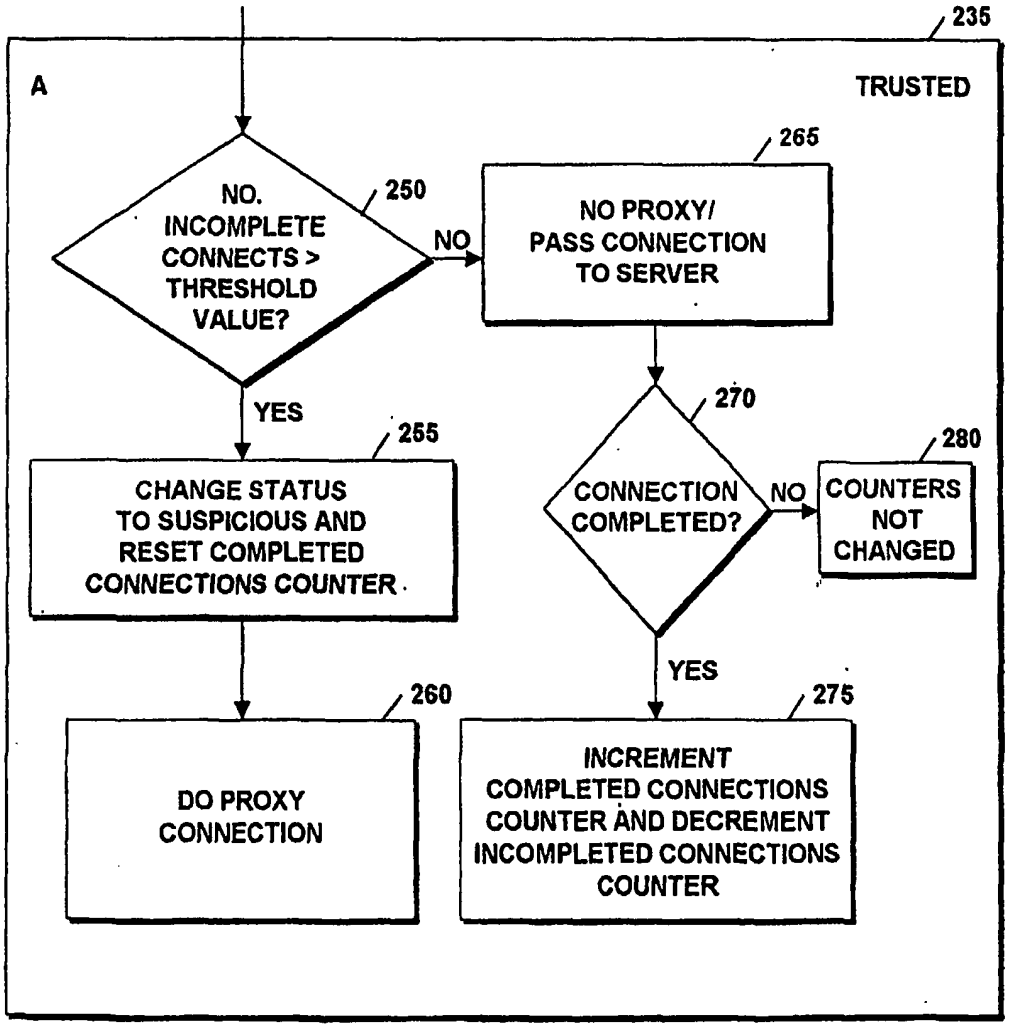

235
A
TRUSTED
NO. INCOMPLETE CONNECTS > THRESHOLD VALUE?
250
NO
265
NO PROXY/ PASS CONNECTION TO SERVER
YES
255
CHANGE STATUS TO SUSPICIOUS AND RESET COMPLETED CONNECTIONS COUNTER
270
CONNECTION COMPLETED?
NO
280
COUNTERS NOT CHANGED
YES
260
DO PROXY CONNECTION
275
INCREMENT COMPLETED CONNECTIONS COUNTER AND DECREMENT INCOMPLETED CONNECTIONS COUNTER

Figure 5

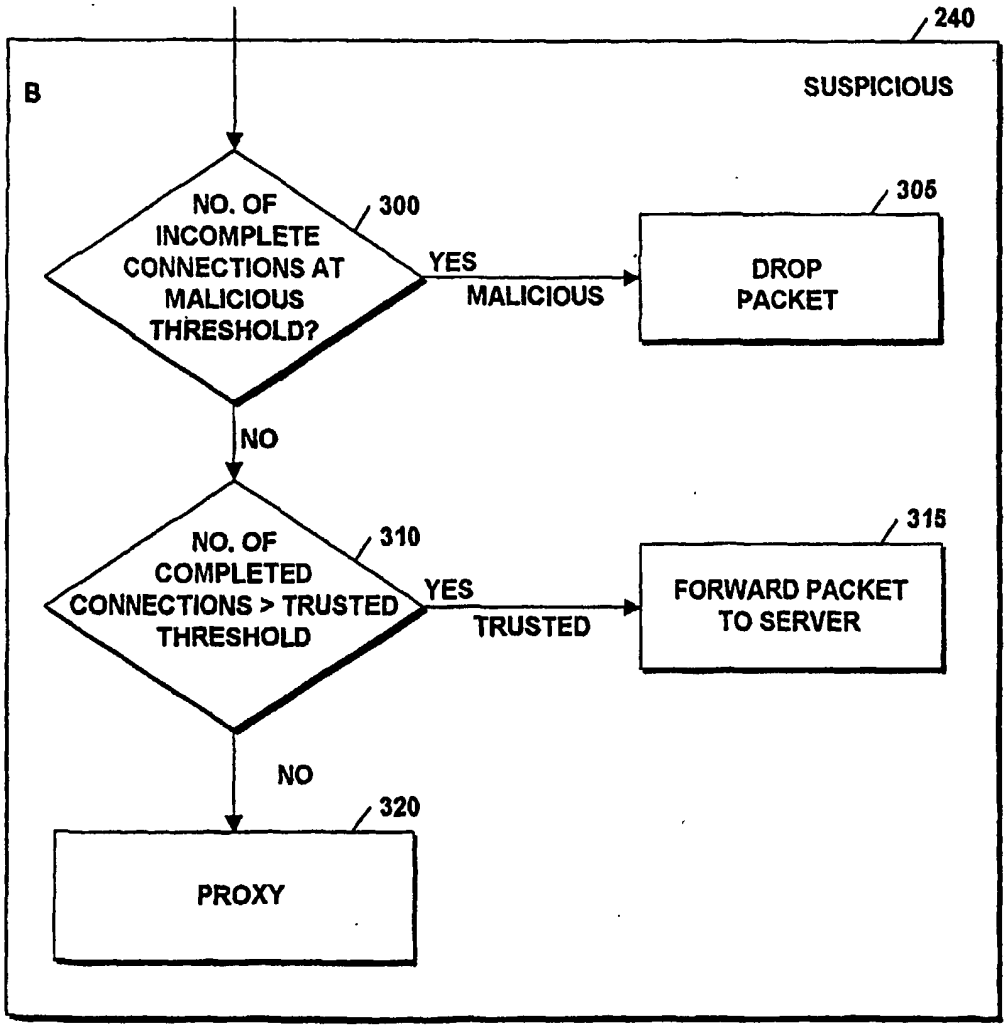

240
B
SUSPICIOUS
NO. OF INCOMPLETE CONNECTIONS AT MALICIOUS THRESHOLD?
300
YES
MALICIOUS
305
DROP PACKET
NO
NO. OF COMPLETED CONNECTIONS > TRUSTED THRESHOLD
310
YES
TRUSTED
315
FORWARD PACKET TO SERVER
NO
320
PROXY

Figure 6

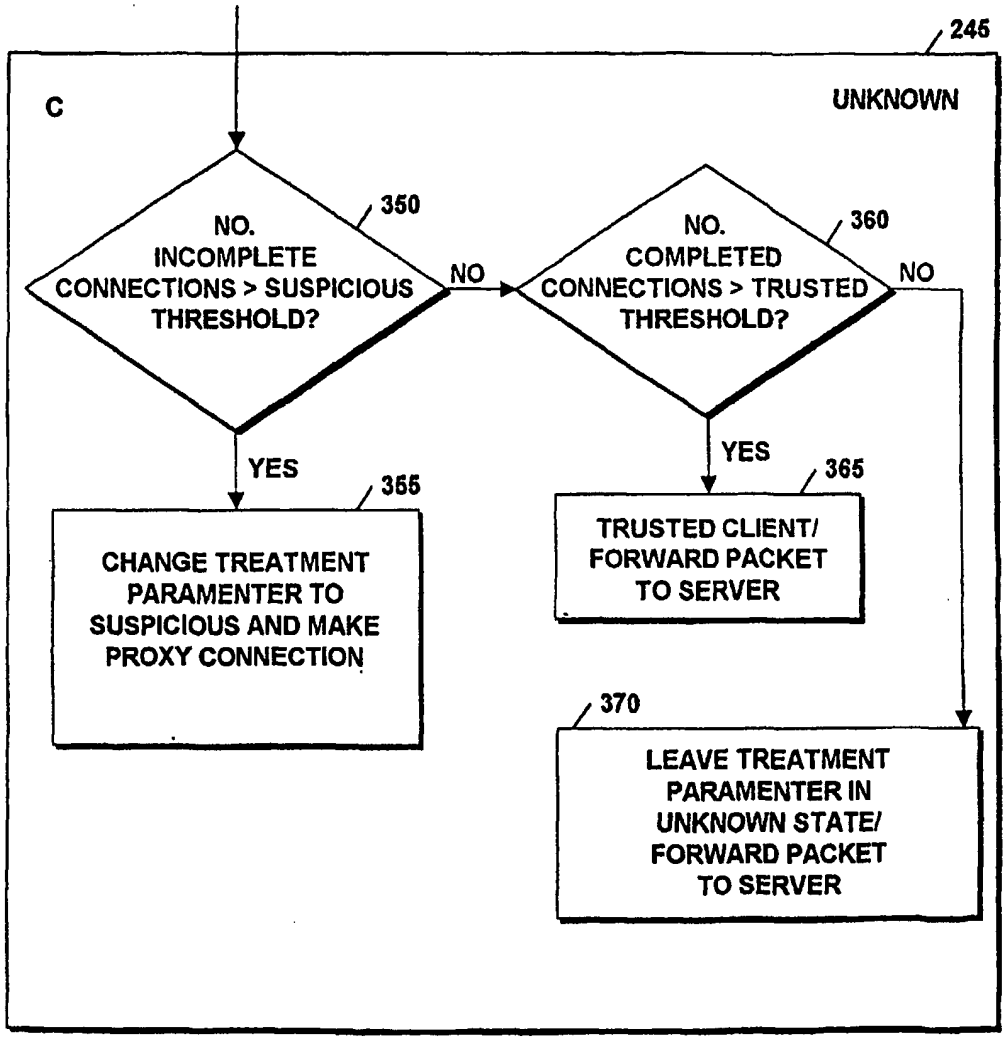

245
C
UNKNOWN
NO.
INCOMPLETE
CONNECTIONS > SUSPICIOUS
THRESHOLD?
350
NO
NO.
COMPLETED
CONNECTIONS > TRUSTED
THRESHOLD?
360
NO
YES
355
CHANGE TREATMENT
PARAMENTER TO
SUSPICIOUS AND MAKE
PROXY CONNECTION
YES
365
TRUSTED CLIENT/
FORWARD PACKET
TO SERVER
370
LEAVE TREATMENT
PARAMENTER IN
UNKNOWN STATE/
FORWARD PACKET
TO SERVER

Figure 7